

LVO3/ 210/508770

10 Rec'd PCT/A

22 SEP 2004

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)



REC'D 28 MAR 2003

WIPO

PCT

LATVIJAS REPUBLIKAS PATENTU VALDE
Patent Office of the Republic of Latvia

APLIECĪBA
Certificate

Pieteikuma Nr.
Application No.

P - 02 - 45

Apliecinām, ka šeit pievienotie dokumenti ir nākamajā lapā aprakstītā Latvijas Republikas Patentu valdē iesniegtā pieteikuma dokumentu precīzas kopijas:

We hereby certify that the attached documents are true copies from the documents of application described on the following page, as originally filed with the Patent Office of the Republic of Latvia

Latvijas Republikas Patentu valdes direktora uzdevumā
For the Director of the Patent Office of the Republic of Latvia

M. Strautzele
Valsts reģistru nodaļas vadītāja
Head of Department of State Registers

Rīgā, 2003. gada 24. martā

BEST AVAILABLE COPY

LATVIJAS REPUBLIKAS PATENTU VALDE

Rīga, Cīradeles iela 7(70)
✉ Pasta adrese:
a/k 824, Rīga, LV 1010 Latvija

Telefons: 7027619
Fax +371 7027690

APLIECĪBAS 2. LAPA
Sheet 2 of the Certificate

IZRAKSTS NO REĢISTRA
Extract of Register

Mūsu šifrs
P-02-45

Mūsu datums
22.03.2002

(21) Pieteikuma numurs:
Application number:

P-02-45

(22) Pieteikuma datums:
Date of filing :

2002. gada 22. marts

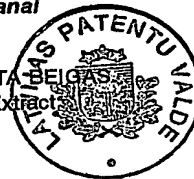
(71) Pieteicējs(i):
Applicant(s):

**DATORU DROŠĪBAS TEHNOLOĢIJA, SIA; Elvīras iela 9c-2,
Rīga LV-1083, LV**

(54) Izgudrojuma nosaukums:
Title of the invention:

**Metode un aparāts datu bezzudumu saspišanai un
atspiešanai**

IZRAKSTA BEIGAS
End of Extract



22.11.2013

22 Saņemšanas datums	20 Saņemšanas numurs P-02-45	21 Pieteikuma numurs
Prioritātes datums	51 ISK indeksi	Trūkstošo dokumentu saņemšanas datums Papildinājumu saņemšanas datums

LATVIJAS REPUBLIKAS
PATENTU VALDEI

IESNIEGUMS
PAR PATENTA IZDOŠANU
IZGUDROJUMAM
PATENT APPLICATION

ADRESE: Citadeles 7 (70), Rīga
☒ a/k 124, Rīga LV - 1010
Tālr.: (371) 702-7344
Fakss: (371) 702-7208

Pieteikuma šifrs
(ja vēlams)

AP/Ap-P658/02

54 Uz pievienoto dokumentu pamata lūdzu izdot patentu izgudrojumam ar nosaukumu:

Metode un aparāts datu bezzudumu saspiešanai un atspiešanai

Patentu lūdzu izdot personai: paredzamais patenta īpašnieks / īpašnieki -
vārds, uzvārds / juridiskās personas nosaukums nominatīvā

dzīves / atrašanās vieta, valsts (kods)

DATORU DROŠĪBAS TEHNOLOĢIJAS, SIA

Elvīras iela 9c-2
Rīga LV-1083 (LV)

Lūdzu noteikt izgudrojuma prioritāti:

☐ Konvencijas prioritāte; dokumenta veids:

31 Dok. Nr.

33 Valsts (kods)

32 Datums

☐ Izstādes / cita prioritāte; dokumenta veids:

23 Datums

72 Izgudrotājs / izgudrotāji:
vārds, uzvārds

dzīves vieta, valsts (kods)

1. Aldis RIGERTS

Ābeļu iela 18
Siguldas pagasts
Rīgas rajons LV-2150 (LV)

2. Valdis ŠĶESTERS

F. Brīvzemnieka iela 10-4
Rīga LV-1004 (LV)

71 Izgudrojuma pieteicējs / pieteicēji:
vārds, uzvārds / juridiskās personas nosaukums nominatīvā

dzīves / atrašanās vieta, valsts (kods)

skat. 73

METODE UN APARĀTS DATU BEZZUDUMU SASPIEŠANAI UN ATSPIEŠANAI

5

Apraksts

Izgudrojums attiecas uz universālajām bezzudumu datu saspiešanas un atspiešanas metodēm, kā arī uz aparātiem to realizēšanai.

10

Datu saspiešanas metodes var tikt raksturotas ar vairākiem galvenajiem parametriem, tādiem kā ātrums, saspiešanas pakāpe, nepieciešamās atmiņas daudzums un realizācijas vienkāršība. Atkarībā no metodes pielietojuma un izvirzītā uzdevuma parasti ir nepieciešams meklēt kompromisu starp minētajiem parametriem. Patreizējā informāciju tehnoloģiju attīstības stadijā aizvien lielāku nozīmi iegūst saspiešanas ātrums, kā arī saspiešanas metožu piemērotība saspiešanai tīklu protokolu līmenī. Šobrīd kā plaši izmantojamas ir zināmas vairākas metodes, piemēram LZx un Predictor metodes.

15

20 LZx datu saspiešanas metožu teorētiskā bāze ir aprakstīta Eibrahema Lempela (Abraham Lempel) un Džeikoba Ziva (Jacob Ziv) darbos, kas publicēti IEEE Transactions on Information Theory, IT-23-3, May 1977, pp. 337-343 un IEEE Transactions on Information Theory, IT-24-5, September 1978, pp. 530-536. Datu saspiešanas un atspiešanas sistēma, kas pazīstama kā LZW sistēma un
25 kura ir adaptēta kā standarts V.42 bis modemos izmantojamajā saspiešanā un atspiešanā, ir aprakstīta, piemēram ASV patentā No. 4,558,302.

Ir zināms, ka LZx metode nodrošina labu saspiešanas pakāpi, bet tai ir relatīvi mazs ātrums, jo tiek izmantotas sarežģītas datu struktūras, kas prasa lielu
30 apstrādes laiku. Tāpat jāatzīmē, ka sarežģīto datu struktūru dēļ ir apgrūtināta šīs metodes realizācija specializētu mikroshēmu veidā un tās pielietošanai ir nepieciešams relatīvi liels atmiņas apjoms.

Metožu ar Predictor izmantošanu teorētiskā bāze ir aprakstīta Timo Raita un Jukka Teuhola darbos "Text compression using prediction," 1986 ACM
35 Conference on Research and Development in Information Retrieval, September 1986 un "Predictive text compression by hashing," The Tenth Annual International ACM SIGIR Conference on Research and Development in

- c) neobligāti, paredzējumu tabulas atjaunināšanu, ierakstot neparedzēto rakstzīmi paredzējumu tajā tabulas šūnā, kura tiek adresēta ar minēto jaucējfunkcijas virkni;
- d) jaucējfunkcijas virknes atjaunināšanu.

5

Bez tam, ja to prasa konkrēta metodes realizācija, solis (b) saskaņā ar šo izgudrojumu papildus var ietvert sekojošus soļus:

- i) minētās neparedzētās rakstzīmes salīdzināšanu ar paredzējumu, kuru satur nākošā paredzējumu tabula;
- 10 ii) nepārtraukti paredzēto rakstzīmju skaita un minētās nākošās paredzējumu tabulas identifikatora kodēšanu, ja minētā neparedzētā rakstzīme sakrīt ar paredzējumu, kuru satur nākošā paredzējumu tabula; vai
- 15 iii) nepārtraukti paredzēto rakstzīmju skaita un uzreiz aiz nepārtraukti paredzētajām rakstzīmēm sekojošās neparedzētās rakstzīmes kodēšanu, ja minētā neparedzētā rakstzīme nesakrīt ar paredzējumu, kuru satur nākošā paredzējumu tabula.

20 Turklāt, gadījumā, kad minētā neparedzētā rakstzīme un minētais paredzējums, kuru satur nākošā paredzējumu tabula, nesakrīt (solis iii), metode saskaņā ar izgudrojumu nodrošina to, ka soļi (i) līdz (iii) var tikt izpildīti rekursīvi, līdz tiek izmantotas visas eksistējošās paredzējumu tabulas.

Iespēja izmantot vairākas paredzējumu tabulas palielina veiksmīgu paredzēšanas gadījumu skaitu un līdz ar to datu saspiešanas pakāpi, kā arī ātrumu.

25

Saspiešanas ātruma palielināšanu ievērojami sekmē arī iespēja ienākošās datu rakstzīmju plūsmas rakstzīmi salīdzināt paralēli ar paredzējumiem, kas atrodas divās vai vairākās eksistējošās paredzējumu tabulās.

30

Lai nodrošinātu labāku adaptāciju saspiežamajiem datiem un līdz ar to vēl vairāk palielinātu to saspiešanas pakāpi, paredzējumu adresācijai dažādās paredzējumu tabulās var tikt izmantotas divas vai vairākas jaucējfunkcijas virknes, pie kam katra jaucējfunkcijas virkne tiek formēta ar unikālas, ienākošajiem datiem korelatīvas jaucējfunkcijas palīdzību.

35

Tālāk, ja nepieciešams piemēroties konkrētiem uzdevumiem, piemēram ātrākai adaptācijai vai lielākai saspiešanas pakāpei, metode saskaņā ar šo izgudrojumu

Fig. 3A un Fig. 3B ir attēlota plūsmkarte, kas ilustrē rakstzīmju secības parauga saspiešanas procesu saskaņā ar pirmo izgudrojuma iemiesto variantu;

Fig. 4 ir attēlota blokshēma, kas ilustrē datu atspiešanas procesu saskaņā ar pirmo izgudrojuma iemiesto variantu;

5 Fig. 5A un Fig. 5B ir parādīta datu struktūra saskaņā ar otro izgudrojuma iemiesto variantu;

Fig. 6A un Fig. 6B ir attēlota blokshēma, kas ilustrē datu saspiešanas procesu saskaņā ar otro izgudrojuma iemiesto variantu.

10 Datu saspiešanas metode saskaņā ar šo izgudrojumu, atkarībā no tādām prasībām kā saspiešanas ātrums, saspiešanas pakāpe, nepieciešamais atmiņas apjoms, izpildījuma vienkāršība, var tikt realizēta vienkāršākos vai sarežģītākos veidos. Lai atvieglotu skaidra un saprotama priekšstata iegūšanu par izgudrojuma būtību, tālāk, atsaucoties uz fig. 1 līdz fig. 4, izklāstīsim vienu
15 no izgudrojuma iemiesto variantiem, kurš ir orientēts uz lielu saspiešanas ātrumu, mazu nepieciešamās atmiņas apjomu un maksimālu vienkāršību.

Šajā izgudrojuma iemiesto variantā jaucējfunkcijas virknes formēšanai no ienākošās datu rakstzīmju plūsmas tiek izmantota viena jaucējfunkcija. Tā kā šī
20 izgudrojumam atbilstošās metodes realizācija ir orientēta uz maksimālu ātrumu un mazu nepieciešamās atmiņas daudzumu, tad tiek izvēlēta vienkārša jaucējfunkcija, kuras virknes garums ir vienāds ar vienas rakstzīmes garumu, bet tās kārtējā vērtība vienmēr ir vienāda ar iepriekš apstrādāto rakstzīmi ($HASH=LastChr$), izņemot procesa sākumu, kad jaucējfunkcijas virknei inicializācijas solī tiek piešķirta vērtība nulle. Fig. 1 ir attēlotas ienākošās
25 rakstzīmes 110, 111 un 112, kuras secīgi apstrādājot, jaucējfunkcijas virknei tiek attiecīgi piešķirtas iepriekš apstrādāto rakstzīmju vērtības 120, 121 un 122.

Jaucējfunkcijas virkne tiek izmantota šūnu adresācijai speciālā atmiņas apgabalā – paredzējumu tabulā 100 (fig. 1). Tā kā šis izgudrojuma iemiesto variants ir orientēts uz lielu saspiešanas ātrumu, mazu nepieciešamās atmiņas apjomu un vienkāršību, tad paredzējumu glabāšanai tiek izmantota viena tabula. Ņemot vērā vēsturiski izveidojušos situāciju, ka datu apstrādes sistēmās vienas rakstzīmes garums parasti tiek pieņemts vienāds ar 8 bitiem, tad arī šajā
30 izgudrojuma iemiesto variantā rakstzīmes garums, kā arī paredzējuma garums tiek pieņemti vienādi ar 8 bitiem. Tādējādi jaucējfunkcijas virkne var pieņemt 256 vērtības un līdz ar to tā spēj adresēt 256 šūnas paredzējumu tabulā. Tas savukārt nosaka paredzējumu tabulai 100 nepieciešamo atmiņas apjomu, kas šajā gadījumā ir tikai 256 baiti.

Detalizēti saspiešanas process ir attēlots fig. 2. Sākumā tiek izpildīts solis 200, kurā jaucējfunkcijas virknei tiek piešķirta vērtība nulle. Šajā solī 200 tiek inicializēta arī paredzējumu tabula, piešķirot tās šūnām vērtību nulle. Tālāk seko soļu kopums 210, kurā tiek iegūts nepārtraukti paredzēto datu rakstzīmju skaits.

- 5 Tas savukārt sastāv no vairākiem soļiem. Solī 211 skaitītājam tiek piešķirta vērtība nulle. Tam seko rakstzīmes ievads solī 212, bet solī 213 tiek pārbaudīts, vai ievadītā rakstzīme sakrīt ar jaucējfunkcijas virknes adresēto paredzējuma vērtību. Ja pārbaudes rezultāts ir pozitīvs, tad solī 214 skaitītāja vērtība tiek palielināta par vienu un saskaņā ar augstāk aprakstīto izmantojamo
- 10 jaucējfunkciju solī 215 tiek atjaunināta jaucējfunkcijas virkne, piešķirot tai apstrādājamās paredzētās rakstzīmes vērtību. Tālāk solī 216 tiek ievadīta nākošā rakstzīme un notiek atgriešanās pie pārbaudes soļa 213. Ja pārbaudes rezultāts solī 213 ir negatīvs, tad solī 220 skaitītāja vērtība tiek kodēta ar Elias
- 15 delta kodu un izvadīta kopā ar neparedzēto rakstzīmi, veidojot saspiesto datu plūsmu. Tad solī 230 tiek atjaunināts paredzējumu tabulas saturs, ierakstot tajā pašā šūnā, ar kuras vērtību notika salīdzināšana, neparedzēto rakstzīmi. Tālāk solī 240 tiek atjaunināta jaucējfunkcijas virkne, piešķirot tai apstrādājamās neparedzētās rakstzīmes vērtību un notiek atgriešanās pie augstāk apskatītā soļa 211. Šādā veidā process tiek cikliski turpināts un rezultātā tiek iegūta
- 20 saspiesta un viennozīmīgi restaurējama datu plūsma.

- Atspiešanas procesā pēc kodēšanai atbilstošas dekodēšanas, apstrādājot skaitītāja vērtību un neparedzēto rakstzīmju plūsmu, tiek iegūta sākotnējā datu plūsma. Atspiešanas process ir attēlots fig. 4. Sākumā tiek izpildīts solis 400,
- 25 kurā tiek inicializēta jaucējfunkcijas virkne un paredzējumu tabula. Šajā solī 400 jaucējfunkcijas virknei un paredzējumu tabulas šūnām tiek piešķirtas tās pašas vērtības, kuras tika izmantotas datu kompresijas sākumā solī 200 (fig. 2) un kas šajā izgudrojuma iemiesojuma variantā ir nulles. Nākošajā solī 410 saspiestā datu plūsma tiek dekodēta un tiek iegūta skaitītāja vērtība un neparedzētā
- 30 rakstzīme. Pēc tam solī 421 tiek pārbaudīts, vai skaitītāja vērtība ir vienāda ar nulli. Ja pārbaudes rezultāts ir pozitīvs, tad solī 430 tiek izvadīta minētā neparedzētā rakstzīme. Tad solī 440 tiek atjaunināts paredzējumu tabulas saturs, ierakstot šo neparedzēto rakstzīmi jaucējfunkcijas virknes adresētajā šūnā. Tālāk solī 450 tiek atjaunināta jaucējfunkcijas virkne, piešķirot tai izvadītās
- 35 neparedzētās rakstzīmes vērtību, un notiek atgriešanās pie augstāk apskatītā soļa 410. Ja pārbaudes rezultāts solī 421 ir negatīvs, tad solī 422 tiek izvadīts jaucējfunkcijas virknes adresētais paredzējums no paredzējumu tabulas. Tad solī 423 tiek atjaunināta jaucējfunkcijas virkne, piešķirot tai solī 422 izvadītā paredzējuma vērtību. Tālāk solī 424 skaitītāja vērtība tiek samazināta par vienu

rakstzīmi ($HASH=LastChr$), izņemot procesa sākumu, kad jaucējfunkcijas virknei inicializācijas solī tiek piešķirta vērtība nulle.

5 Tātad, apstrādājot konkrētu ienākošo rakstzīmi, kārtējās jaucējfunkciju virknes adresē paredzējumus atbilstošās paredzējumu tabulās. Kā parādīts fig. 5A, jaucējfunkcijas virkne 1 (520) adresē paredzējumu 1 (511), paredzējumu 2 (512) un paredzējumu 3 (513), bet jaucējfunkcijas virkne 2 (521) adresē paredzējumu 4 (514). Tā kā rakstzīmes garums apskatāmajā izgudrojuma iemiesojuma variantā ir pieņemts vienāds ar 8 bitiem, tad jaucējfunkcijas virknes 1 (520) garums ir 15 biti, bet jaucējfunkcijas virknes 2 (521) garums ir 8 biti. Tas savukārt nosaka paredzējumu tabulu apjomu, proti paredzējumu tabulām 1, 2 un 3 nepieciešami 32 kilobaiti, bet paredzējumu tabulai 4 – 256 baiti.

15 Vairāku dažādu jaucējfunkciju kombinēta izmantošana nodrošina to, ka pieaug adaptācijas ātrums un piemērotība dažādiem datu tipiem. Rezultātā tas ļauj iegūt augstāku datu plūsmas saspiešanas pakāpi.

20 Detalizēti saspiešanas process pa soļiem ir parādīts fig. 6A un fig. 6B. Sākumā tiek izpildīts inicializācijas solis 600, kurā jaucējfunkciju virknēm tiek piešķirtas nulles vērtības. Šajā solī 600 tiek inicializētas arī paredzējumu tabulas, aizpildot to šūnas ar nullēm. Tālāk solī 610 seko rakstzīmes ievads un solī 620 tiek pārbaudīts, vai jaucējfunkcijas virknes 1 adresētie paredzējums 1 un paredzējums 2 ir vienādi ar nulli. Ja pārbaudes rezultāts solī 620 ir negatīvs, 25 proti, ja kaut viens no pārbaudītajiem paredzējumiem nav vienāds ar nulli, tad tiek izpildīts soļu kopums 630, kurā ar skaitītāja palīdzību tiek iegūts nepārtraukti paredzēto ienākošo datu rakstzīmju skaits. Soļu kopums 630 sākas ar soli 631, kurā skaitītājam tiek piešķirta vērtība nulle. Tālāk solī 632 tiek pārbaudīts, vai apstrādājamā rakstzīme sakrīt ar jaucējfunkcijas virknes 1 adresēto 30 paredzējumu 1. Ja pārbaudes rezultātā tiek konstatēts, ka apstrādājamā rakstzīme sakrīt ar paredzējumu 1, tad tiek izpildīts solis 633, kurā skaitītāja vērtība tiek palielināta par vienu. Tad solī 634, atbilstoši izmantojot augstāk aprakstītās jaucējfunkcijas, tiek atjauninātas jaucējfunkcijas virknes 1 un jaucējfunkcijas virknes 2 vērtības. Tālāk solī 635 seko nākošās rakstzīmes 35 ievads no ienākošās datu plūsmas un notiek atgriešanās pie soļa 632. Tādā veidā skaitītājā tiek uzkrāts nepārtraukti paredzēto datu rakstzīmju skaits.

Ja pārbaudes solī 632 tiek konstatēts, ka apstrādājamā rakstzīme nesakrīt ar paredzējumu 1, tad tiek izpildīts solis 650, kurā tiek pārbaudīts, vai skaitītāja

kurā, atbilstoši izmantojot augstāk aprakstītās jaucējfunkcijas, tiek atjauninātas jaucējfunkcijas virkne 1 un jaucējfunkcijas virkne 2 un notiek atgriešanās pie rakstzīmes ievades soļa 610.

- 5 Ja solī 660 veiktās pārbaudes rezultāts ir negatīvs, tad tiek izpildīts solis 670, kurā tiek pārbaudīts, vai apstrādājamā rakstzīme sakrīt ar jaucējfunkcijas virknes 1 adresēto paredzējumu 3. Ja solī 670 veiktās pārbaudes rezultāts ir pozitīvs, tad tiek izpildīts solis 671, kurā tiek atjaunināts paredzējumu tabulas 2 un paredzējumu tabulas 3 saturs. Atjaunināšanas procesā ar jaucējfunkcijas virkni 1 adresētais paredzējums 2 no paredzējumu tabulas 2 tiek pārņemts uz paredzējumu tabulas 3 šūnu, kuru adresē jaucējfunkcijas virkne 1. Savukārt paredzējumu tabulā 2 jaucējfunkcijas virknes 1 adresētajā šūnā tiek ierakstīta apstrādājamā rakstzīme. Rezultātā jaucējfunkcijas virknes 1 adresētās paredzējuma 2 un paredzējuma 3 vērtības tiek savstarpēji samainītas vietām.
- 10 Tālāk tiek izpildīts solis 672, kurā tiek kodēts un izvadīts kods $259 + \text{skaitītājs} * 3$. Tam seko solis 699, kurā, atbilstoši izmantojot augstāk aprakstītās jaucējfunkcijas, tiek atjauninātas jaucējfunkcijas virkne 1 un jaucējfunkcijas virkne 2 un notiek atgriešanās pie rakstzīmes ievades soļa 610.
- 15

- 20 Ja solī 670 veiktās pārbaudes rezultāts ir negatīvs, tad tiek izpildīts solis 680, kurā tiek kodēts un izvadīts kods $260 + \text{skaitītājs} * 3$. Tālāk seko jau augstāk aprakstītie solis 690 un aiz tā sekojošie tālākie soļi.

- Šeit saspiešanas procesa aprakstā atgriežamies pie jau augstāk apskatītā soļa 620, kurā tiek pārbaudīts, vai jaucējfunkcijas virknes 1 adresētie paredzējums 1 un paredzējums 2 ir vienādi ar nulli. Ja solī 620 veiktās pārbaudes rezultāts ir pozitīvs, tad tiek izpildīts solis 640, kurā paredzējumu tabulas 1, paredzējumu tabulas 2 un paredzējumu tabulas 3 saturs tiek atjaunināts tādā veidā, ka jaucējfunkcijas virknes 1 adresētajiem paredzējumam 1 tiek piešķirta apstrādājamās rakstzīmes vērtība, paredzējumam 2 tiek piešķirta rakstzīmes ' ' (20H) vērtība un paredzējumam 3 tiek piešķirta rakstzīmes 'e' (65H) vērtība.
- 30 Tālāk seko jau augstāk aprakstītie solis 690 un aiz tā sekojošie tālākie soļi. Šādā veidā paredzējumu tabulu šūnām, kuras pēc inicializācijas procesa sākumā tālākajā saspiešanas procesā vēl nav aizpildītas, tiek piešķirtas iepriekš noteiktu potenciāli biežāk sastopamo rakstzīmju vērtības, kas palielina
- 35 paredzēto rakstzīmju skaitu un uzlabo datu plūsmas saspiešanas pakāpi.

Saspiešanas procesa soļos 651, 662, 672, 680, 691, 694 tiek kodēta un izvadīta informācija, veidojot saspiesto datu plūsmu. Tā kā rakstzīmes pieņemtais

bez paredzējumu tabulu atjaunināšanas. Tādā gadījumā paredzējumu tabulu saturs tiek sagatavots iepriekš un saspiešanas procesā netiek atjaunots.

- 5 Tālāk, iespējams realizācijas variants, kad saspiešanas procesa sākumā paredzējumu tabulas tiek inicializētas ar iepriekš sagatavotu saturu, kas saspiešanas procesa laikā tiek atjaunināts. Šāds realizācijas variants var būt efektīvs, kad iepriekš ir daļēji zināma saspiežamo datu struktūra.

- 10 Ja tiek izmantotas vairākas paredzējumu tabulas, tad iespējami varianti, kad dažādām paredzējumu tabulām tiek izmantotas dažādas atjaunināšanas stratēģijas. Šādu kombinētu stratēģiju pielietošana ļauj piemērot datu saspiešanas metodi saskaņā ar izgudrojumu visdažādākajiem mērķiem. Apskatītajā otrajā izgudrojuma iemiesojuma variantā paredzējumu tabulu atjaunināšanas procesā paredzējumu vērtības tiek arī pārvietotas starp tabulām,
- 15 kā rezultātā saspiešanas procesā biežāk sastopamās vērtības nonāk paredzējumu tabulās, ar kurām apstrādājamās rakstzīmes salīdzināšana notiek vispirms. Arī šādos gadījumos iespējamās dažādas stratēģijas, kas speciālistam nesagādās grūtības tās izmantot pēc vajadzības konkrētu mērķu sasniegšanai.

- 20 Tālāk, saskaņā ar izgudrojumu var tikt izmantotas vairākas atšķirīgas jaucējfunkcijas, kuru kombinēta izmantošana nodrošina to, ka pieaug adaptācijas ātrums un piemērotība dažādiem saspiežamo datu tipiem. Tas rezultātā ļauj iegūt augstāku kompresijas pakāpi.

- 25 Apskatītajos izgudrojuma iemiesojuma variantos soļi tiek izpildīti virknē, bet ir iespējami metodes realizācijas varianti, kad vismaz daļa no soļiem tiek izpildīti paralēli. Piemēram, paralēli iespējams salīdzināt tekošo apstrādājamo rakstzīmi ar vairākiem paredzējumiem, kas rezultātā dod lielāku kompresijas ātrumu.

- 30 Metode saskaņā ar izgudrojumu var tikt izmantota gan atsevišķi, gan arī kopā ar citām metodēm. Bez tam, jāatzīmē metodes izpildes laika augstā prognozējamības pakāpe, proti iespējamās tās realizācijas, kurās, neatkarīgi no ienākošās datu plūsmas rakstura, tās apstrādes laiks ir prognozējams un var tikt arī fiksēts.

Izgudrojuma formula

1. Metode ienākošās datu rakstzīmju plūsmas kodēšanai saspiestas kodu izejas plūsmas iegūšanai, kas ietver sekojošus soļus:

- a) nepārtraukti paredzēto rakstzīmju skaitīšanu, salīdzinot ienākošās datu rakstzīmju plūsmas rakstzīmi ar paredzējumu, kas atrodas paredzējumu tabulā un ir adresēts ar jaucējfunkcijas virkni, pie kam minētā paredzējumu tabula satur lielu skaitu paredzējumu, minētie paredzējumi ir ienākošās datu plūsmas rakstzīmes un/vai iepriekš noteiktas vērtības, minētās jaucējfunkcijas virknes tiek formētas ar ienākošajiem datiem korelatīvas jaucējfunkcijas palīdzību;
- b) nepārtraukti paredzēto rakstzīmju skaita un uzreiz aiz nepārtraukti paredzētajām rakstzīmēm sekojošās neparedzētās rakstzīmes kodēšanu;
- c) neobligāti, paredzējumu tabulas atjaunināšanu, ierakstot neparedzēto rakstzīmi paredzējumu tabulas šūnā, kura tiek adresēta ar minēto jaucējfunkcijas virkni;
- d) jaucējfunkcijas virknes atjaunināšanu.

2. Metode saskaņā ar 1. punktu, kur solis (b) papildus ietver sekojošus soļus:

- i) minētās neparedzētās rakstzīmes salīdzināšanu ar paredzējumu, kuru satur nākošā paredzējumu tabula;
- ii) nepārtraukti paredzēto rakstzīmju skaita un minētās nākošās paredzējumu tabulas identifikatora kodēšanu, ja minētā neparedzētā rakstzīme sakrīt ar paredzējumu, kuru satur nākošā paredzējumu tabula; vai
- iii) nepārtraukti paredzēto rakstzīmju skaita un uzreiz aiz secīgajām paredzētajām rakstzīmēm sekojošās neparedzētās rakstzīmes kodēšanu, ja minētā neparedzētā rakstzīme nesakrīt ar paredzējumu, kuru satur nākošā paredzējumu tabula.

3. Metode saskaņā ar 2. punktu, kur minētās neparedzētās rakstzīmes un minētā paredzējuma, kuru satur nākošā paredzējumu tabula, nesakrišanas gadījumā soļi (i) līdz (iii) tiek izpildīti rekursīvi, līdz tiek izmantotas visas eksistējošās paredzējumu tabulas.

d) jaucējfunkciju virknes atjaunināšanai.

10. Aparāts saskaņā ar 9. punktu, kur līdzekļi (b) papildus satur līdzekļus, kas ir piemēroti:

5

i) minētās neparedzētās rakstzīmes salīdzināšanai ar paredzējumu, kuru satur nākošā paredzējumu tabula;

10

ii) nepārtraukti paredzēto rakstzīmju skaita un minētās nākošās paredzējumu tabulas identifikatora kodēšanai, ja minētā neparedzētā rakstzīme sakrīt ar paredzējumu, kuru satur nākošā paredzējumu tabula; vai

15

iii) nepārtraukti paredzēto rakstzīmju skaita un uzreiz aiz nepārtraukti paredzētajām rakstzīmēm sekojošās neparedzētās rakstzīmes kodēšanai, ja minētā neparedzētā rakstzīme nesakrīt ar paredzējumu, kuru satur nākošā paredzējumu tabula.

20

11. Aparāts saskaņā ar 10. punktu, pie kam minētie līdzekļi ir pielāgoti, lai minētās neparedzētās rakstzīmes un minētā paredzējuma, kuru satur nākošā paredzējumu tabula, nesakrišanas gadījumā soļi (i) līdz (iii) varētu tikt pielietoti rekursīvi, līdz tiek izmantotas visas eksistējošās paredzējumu tabulas.

25

12. Aparāts saskaņā ar 10. vai 11. punktu, pie kam minētie līdzekļi ir pielāgoti, lai paredzējumu adresācijai dažādās paredzējumu tabulās varētu tikt izmantotas divas vai vairākas jaucējfunkcijas virknes, pie kam katra jaucējfunkcijas virkne tiek formēta ar unikālas, ienākošajiem datiem korelatīvas jaucējfunkcijas palīdzību.

30

13. Aparāts saskaņā ar 10. vai 11. punktu, kas neobligāti satur līdzekļus paredzējumu tabulu atjaunināšanai saskaņā ar iepriekš noteiktu stratēģiju.

35

14. Aparāts saskaņā ar 10. vai 11. punktu, kurš papildus satur līdzekļus ienākošās datu plūsmas rakstzīmes paralēlai salīdzināšanai ar paredzējumiem, kas atrodas divās vai vairākās eksistējošās paredzējumu tabulās.

15. Aparāts saskaņā ar 9. punktu, kurš papildus satur līdzekļus, kas ir piemēroti, lai procesa sākumā inicializētu minēto jaucējfunkcijas virkni, piešķirot tai iepriekš noteiktu vērtību.

Kopsavilkums

Izgudrojums attiecas uz metodi ienākošās datu rakstzīmju plūsmas kodēšanai, lai iegūtu saspiestu kodu izejas plūsmu. Metode ir balstīta uz apstrādājamās datu plūsmas rakstzīmju paredzēšanu, salīdzinot tās ar paredzējumiem vienā vai vairākās paredzējumu tabulās un uzskaitot nepārtraukti paredzētās rakstzīmes, tādējādi būtiski samazinot izvadišanas operāciju skaitu. Adresācija paredzējumu tabulās tiek realizēta ar vienas vai vairāku jaucējfunkciju virkņu palīdzību, no kurām katra tiek formēta ar unikālas, ienākošajiem datiem korelatīvas jaucējfunkcijas palīdzību. Šāda veida datu plūsmas apstrāde ļauj novērst no pieņemtā rakstzīmes garuma atkarīgo saspiešanas pakāpes ierobežojumu, tādējādi paaugstinot saspiešanas pakāpi, un vienlaicīgi arī ievērojami palielināt datu apstrādes ātrumu. Izgudrojums attiecas arī uz saspiešanai adekvātu atspiešanas metodi, kā arī uz aparātiem to realizācijai.

Fig. 2

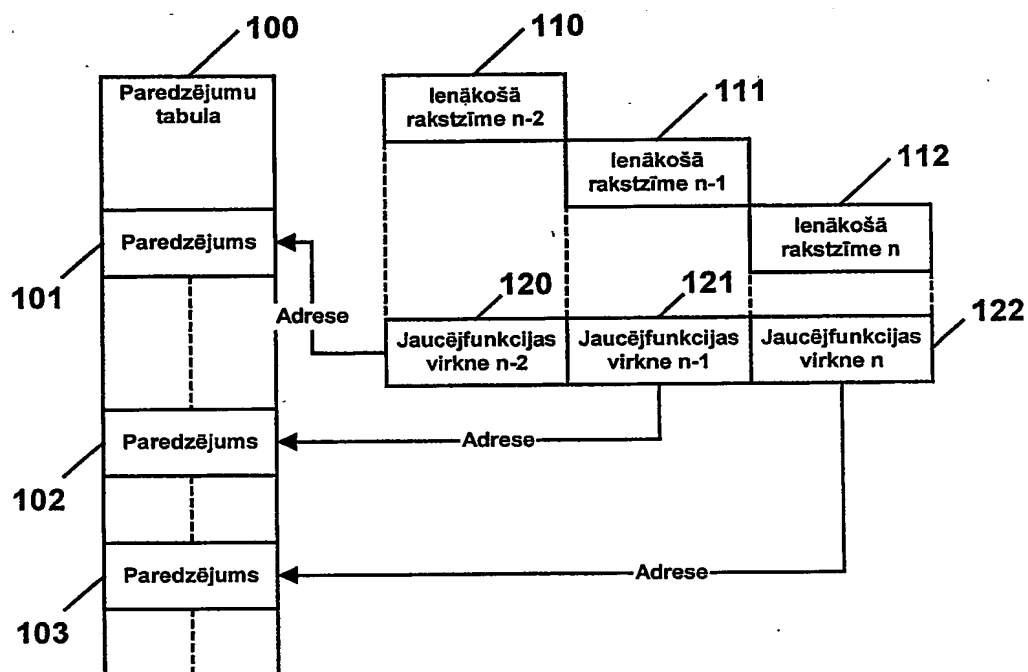


Fig.1

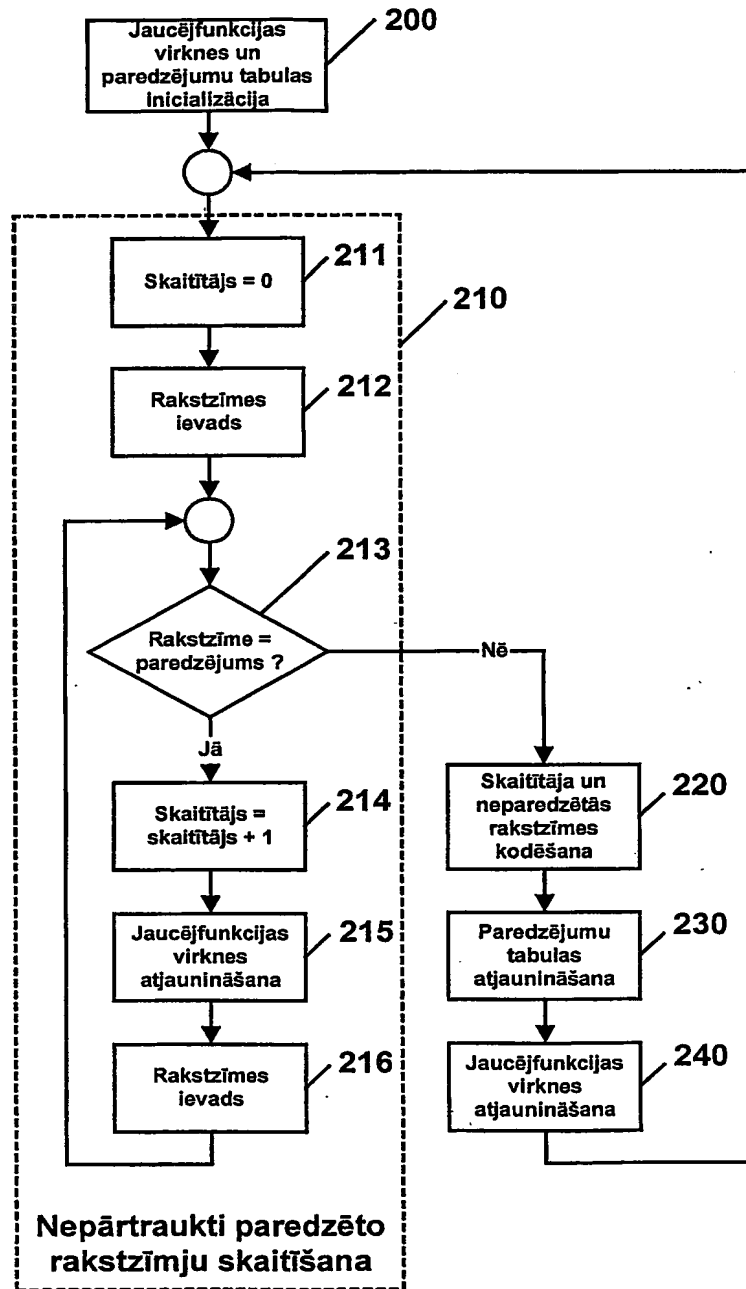


Fig. 2

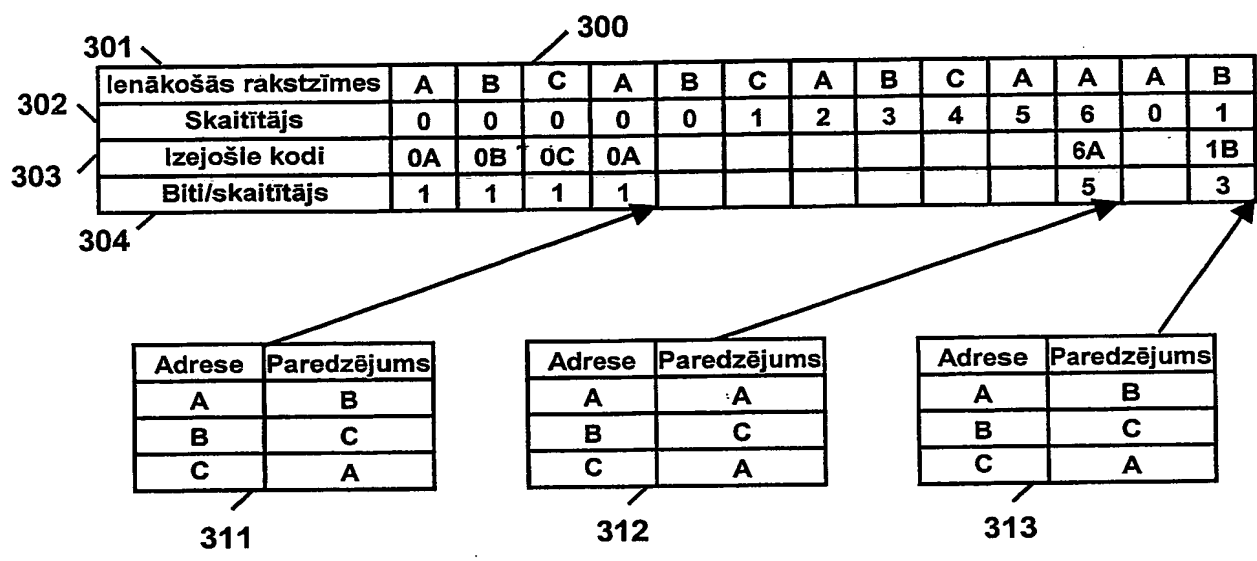


Fig. 3A

Skaitītājs	Kods
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111

Fig. 3B

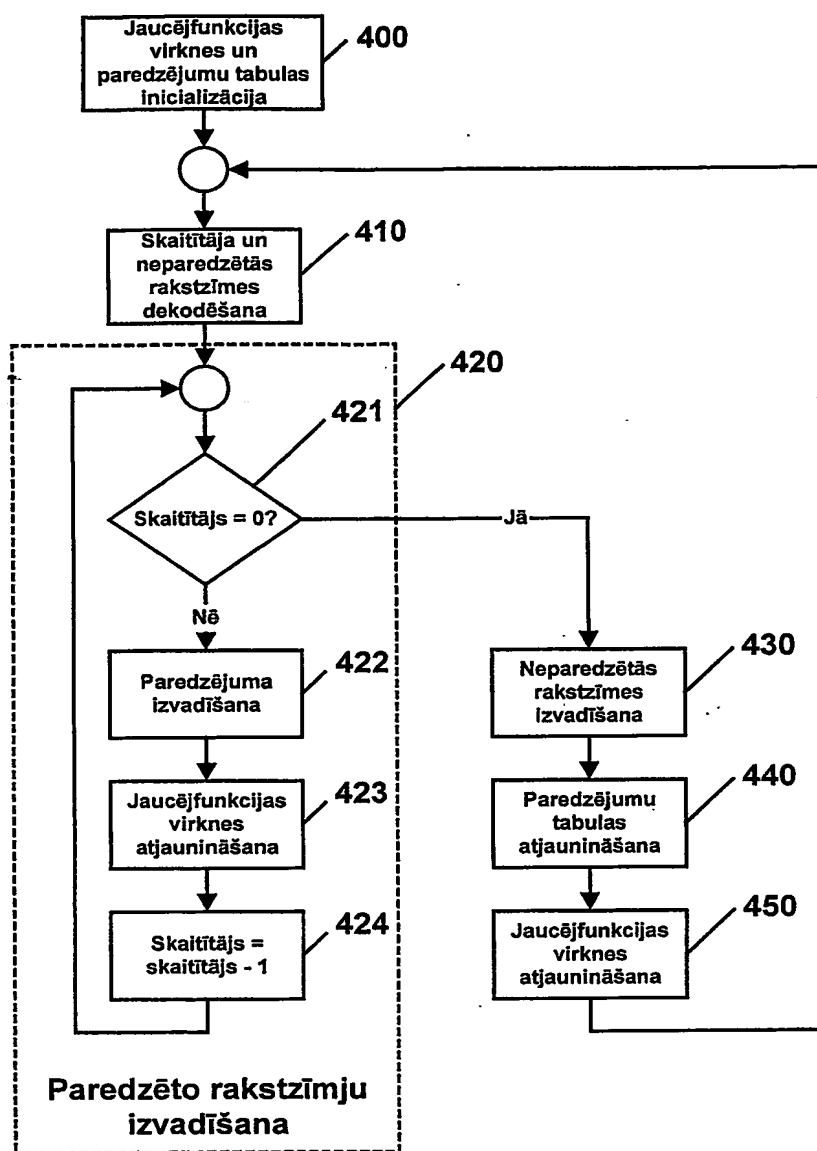


Fig. 4

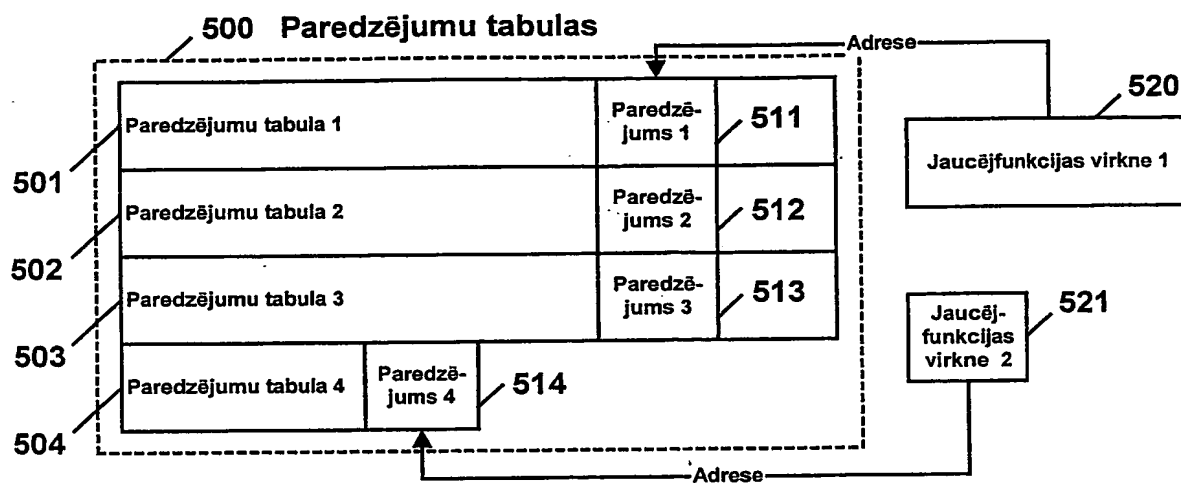


Fig. 5A

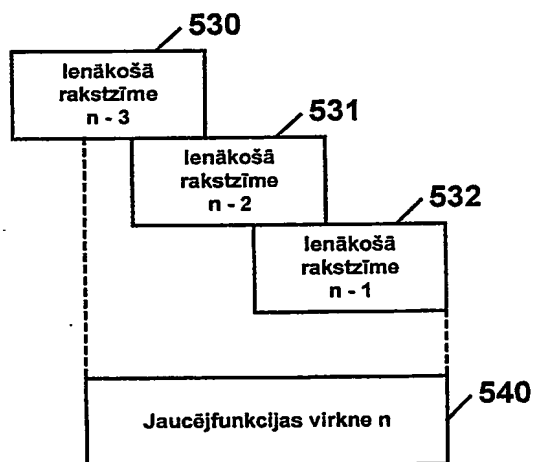
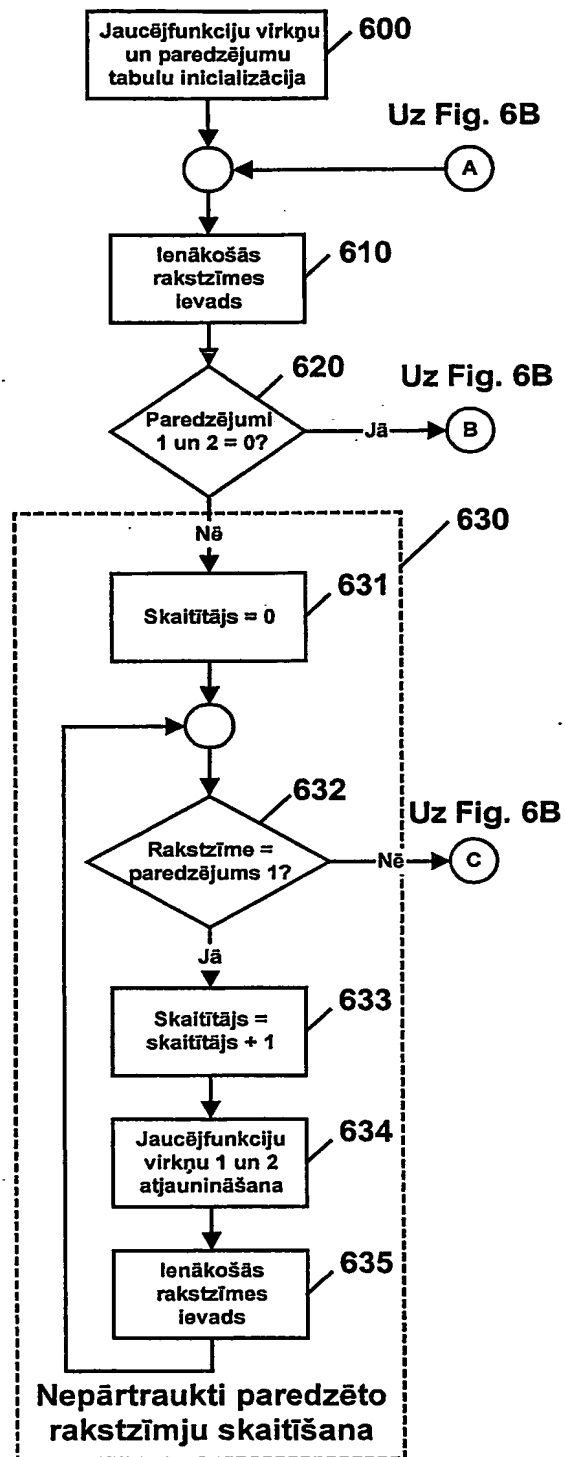
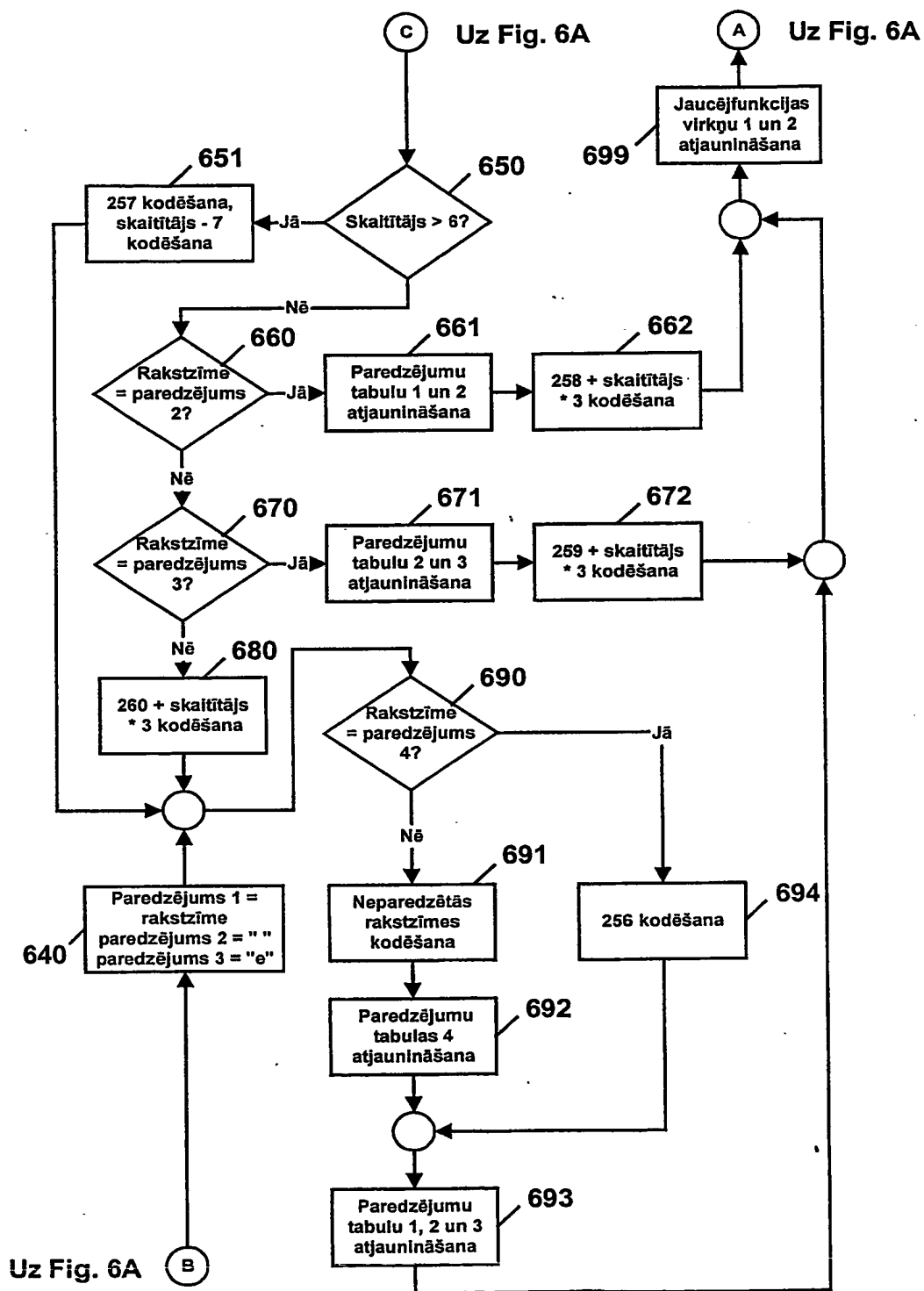


Fig. 5B





**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.